UNCLASSIFIED

AD 271 997

Reproduced by the

UNCLASSIFIED

UBG-61-230/B

Copy 192 of 250

Institute for Defense Analyses

Research and Engineering Support Division

Technical Report Number

61-12

COMPUTERS IN COMMAND AND CONTROL

NOVEMBER 1961

TR 61-12

COMPUTERS IN COMMAND AND CONTROL

NOVEMBER 1961

Institute for Defense Analyses
Research and Engineering Support Division

Contract SD-50

Prepared for the Office of the Assistant Director of Defense Research and Engineering (Communications) as Task 17 Advanced Research Projects Agency Contract No. SD-50 to the Institute for Defense Analyses by:

Dr. Thomas G. Belden	Institute for Defense Analyses, Research and Engineering Support Division
Mr. Robert Bosak	System Development Corporation
Mr. William L. Chadwell	Defense Communications Agency
Dr. Lee S. Christie	System Development Corporation
Mr. John P. Haverty	The Rand Corporation
Dr. E.J. McCluskey, Jr.	Princeton University
Mr. Robert H. Scherer	Office Assistant Director of Defense Research and Engineering for Communications
Dr. Warren S. Torgerson	Lincoln Laboratory, MIT

Marlin G. Kroger, Chairman
Digital Computer Application Study
Institute for Defense Analyses
Research and Engineering Support Division

APPROVED:

Dr. John F. Kincaid, Director
Institute for Defense Analyses
Research and Engineering Support Division

CONTENTS

PART I

INTRODUCTION

On 27 June 1961, at the request of the Director of Defense Research and Engineering (DDR&E), the Advanced Research Projects Agency (ARPA) assigned a task entitled "Digital Computer Application Study" to the Research and Engineering Support Division (RESD) of the Institute for Defense Analyses (IDA).

The following objectives were specified:

1. To study command and control problems with a view toward determining criteria for the effective application of computers to command and control.
2. To postulate goals for future DOD growth in automated command and control capability and to generate guidelines which will aid future planners in specifying individual system characteristics which lend themselves to internetting -- both within the individual systems and particularly within the over-all DOD command and control system.
3. To delineate problem areas needing accelerated research.

The most significant findings of the study group are:

1. The current technical feasibility of using computers to process information in the abstract ways necessary to carry out command functions is commonly overrated. The complexity of the command information processing problem is such that the personnel of the command must still be the dominant information processing elements of the system. Computers are primarily useful for such operational decision-supporting functions as information storage, retrieval, and display. Computers can also be of considerable assistance in the development, evaluation, and modification of plans and in the assessment of force capability.

2. Analyzing and understanding the information needs of a command and developing an appropriate system growth pattern are much more important to the early and continuing success of an automated command information system than are such matters as the choice of a particular computer, e.g., CDC 1604 versus IBM 7090.

3. Command personnel are an integral element of any command system. As such, they should be intimately involved in the on-going activity of design for their computer-aided command information processing system. The responsibility for controlling system evolution cannot be delegated successfully to a development agency outside the command, for several reasons. First, there is a danger that the command will depend on automated decision aids without realizing the extent to which human judgment of operational parameters has been built into such aids by the outside developer. Second, an outside agency will lack an essential system building block -- the command personnel themselves. Third, such an agency will fall well short of the intimate understanding that the command itself has of its functions and problems. Fourth, the problems and necessary functions of the command change unpredictably at such a rate that guiding system development by the transmittal of relatively fixed requirements is ineffective.

4. Each operational command, particularly in the area of joint operations, needs a sharp increase in technical capability in order to control the evolution of its information processing system to assure meeting its specific needs. Single service systems typically have "user representatives" involved in system development; joint command systems often lack even this mechanism. Increased capability is needed on a broad front to strengthen planning, exercising, and evaluation in intimate association with automation. In part, this capability must be acquired within the command line, but substantial technical assistance for analysis, design, and implementation efforts in evolving the information system is also required. This technical assistance must have a close, two-way working relationship with the command at all levels, particularly at the top. It must be accorded a responsible role in the exercise of independent technical judgment.

5. There is a lack of coordination between individual Service automation efforts, and often between the Services and the Unified and Specified Commands, resulting in technical and functional incompatibilities in various complexes of systems. These incompatibilities prevent realization of the full, improved effectiveness that could be afforded by automation.

6. The state-of-the-art in machine language and programming language is adequately advanced so that standards could be established without impeding further research. The present lack of such standards is a significant factor in intra- and inter-system compatibility problems.

7. The state-of-the-art in information techniques, such as problem formulation, analysis, modeling, and design and command languages, is the primary technical factor limiting our capability to apply automation to command systems.

The following sections of this report develop the arguments and discussions which support the findings summarized above and the recommendations in Part VI. The Appendix contains a listing of briefings and source documents contributing to the report.

PART II

BACKGROUND OF THE PROBLEM

A. The Command and Control Problem

The problem of command and control of military forces and resources is as old as war itself. Modern weapons and delivery systems have made the problem more complex and critical. The inappropriate use of even a small tactical nuclear weapon could trigger an escalated response resulting in an unncessary "all out" nuclear war. Considerations such as this, coupled with concern about the shortness of the time it would now take to deliver a highly destructive attack, cause persons in positions of responsibility to want increased control of the actions and reactions of the nation's Armed Forces.

The term "command and control" has become popular and is used to describe a general capability relating to the direction of Armed Forces. Computers are now thought of as integral parts of modern command and control systems, and there is no question about the improved performance they potentially offer. However, the uses to which computers can be put vary considerably with the type of command and control problem.

B. Differences Between Command Functions and Control Functions

Even though not all systems can be neatly categorized as either command systems or control systems because most have elements of both, it is helpful in examining the problem to differentiate between the command functions and control functions.

One basic differentiation which can be made is that command functions involve broad problems of planning, assessing the capabilities of the command's forces and those of the enemy, allocating resources, alerting, and committing the command's forces, etc. These functions require the gathering of large amounts and many classes of information, aggregating the information, and processing it to enable a commander to make knowledgeable, deliberate decisions in a context of changing objectives. Control functions, on the other hand, characteristically involve direct control of weapons in situations where, although the volume of information is large, it can be categorized in a relatively few classes. Objectives are fixed and the problem is to maintain action toward the objectives through error detection and corrective action. The operation of an air defense direction center is a typical control function because the system elements have parameters which can be reduced to specific values, and both the rules for the employment of system elements and the relationships among the elements are well understood.

With these distinctions between command and control in mind, we observe that there is a shift in emphasis between command functions and control functions at various levels of command. At lower levels of command, for example, a SAGE direction center, the control function is a dominant factor. On the other hand, at higher levels, the command function dominates. The cross-over point occurs at different echelons depending on command mission -- at Division level in the Field Army case, at Specified Command level in the SAC case.

This study group gave most of its attention to the command problem since the application of computers to command systems poses a more significant current problem.

C. The Command Function

The command cycle starts by the commander asking himself, "What is my mission"? His mission constitutes his basic linkage to the next higher authority and the policies, directives, orders, and commands imposed on him. The second step is to ask, "What is the status of my forces (and other friendly forces) and the enemy forces, together with political and physical environmental forces which may affect status"? The third step is to determine his alternative courses of action -- either posturing or in combat -- as opposed to the courses of action open to the enemy. The fourth step is to develop a set of plans covering feasible alternatives. Those plans range from those dealing with several contingencies down to a single plan of operation. From the selected operation plan, the commander issues orders, and finally a command is given to execute the order. The commander then follows up to see how the execution of the order compares with his intentions, and the result may generate a new planning-to-command cycle.

There are times when a commander must abridge, modify, or adjust the command process due to the press of time or new situations which must be encompassed in his plans.

D. The Command Environment

The environment of command systems also affects the type of information needed by the commander. Examples of this environment are: the type of conflict (cold war, limited war, general war); the phase of conflict (tension, potential warning, exchange, recovery); the physical location (fixed, mobile); the doctrine (scope of command, succession of command); the support conditions (communications, logistics).

The design, implementation, and operation of information systems can be deeply influenced by these factors of the command environment. For example, in tension situations, very high levels of command may require very detailed pieces of information which, under other environmental conditions, might be trivial. In one situation, the information system may be required to analyze large volumes of information and in another, only the most critical items in the shortest possible time.

E. The Potential Role of Computers

To establish a perspective for examining the potential of computers in command systems, a comparison with the game of chess is suggested. Chess is very much simpler than a high-level military command problem. In chess, the rules are fixed and are the same for both opponents, the intelligence and situation display of the size and deployment of forces is perfect; pawns do not run out of fuel or ammunition; communications is not a problem, etc. Yet, C.E. Shannon has calculated that there are 10^{120} possible plays of the game -- which means that it would take impossibly long to play a game of chess if all alternatives were searched in order to choose an optimum strategy. This is true even if the biggest and fastest conceivable computer were used. However, chess-playing computer programs have been written which allow a computer to play a fair game of chess. Only the more likely moves are examined as the game progresses and they are only examined for a few moves ahead. It is evident that quite good programs could be written to aid a human chess player in making sure that his planned moves would not result in some serious loss through oversight. Military commands could undoubtedly benefit from similar assistance in plan evaluation. Unfortunately it is not always easy to implement such a capability.

PART III

COMMAND SYSTEM INTEGRATION AND EVOLUTION

A. Operational Command Responsibility in Automation

The introduction of current computer capability into command systems will not significantly decrease our dependence on the commander and his staff for information processing. The computer system will provide increased capability to present organized information and can aid in some functions such as planning, war gaming, etc. Since the computer system will be embedded in the operation, the language, and the mission of the particular command, it is important to recognize that the information structure, the data base, computer programs, etc., will be unique to the particular command requiring the system.

Because there are several Unified and Specified Commands and several levels of command in each, someone is likely to propose we design an all-purpose configuration. We question whether one can profitably think of a generalized command system. There are such striking differences between commands that, while standardization of hardware modules seems feasible and some techniques generated for one system will be effective in others, the system organization and the operational program must be tailored to each command. DOD compatibility constraints must ensure hardware and language compatibility, but should not infringe further on the command prerogatives in system organization.

Another factor bearing on command system automation is that the automation of the system must continue to grow, adapt and change throughout the history of the command. This process must be controlled by the command. There is danger of too much dependence on organizations outside the command for decisions relating to the performance parameters, interpretation of mission, and the data base that go into a system model. This "delegation" of command responsibility can constitute a usurpation (or an acquisition through default) of the prerogatives of operational command -- unless the commander is intimately aware of all significant judgment decisions that have been made in the writing of the system program.

It is our assessment that most of the commands do not have sufficient technical capability to undertake the responsibilities which we have outlined above. Therefore, most commands will need additional trained technical personnel on their staffs in order to effectively manage the evolutionary design and implementation of the command system. These officers will be required as important links in the design process of determining what the commander and his staff need and what is technically achievable.

We recognize that it will be necessary to secure outside technical assistance. Such technical help must have a two-way communication channel with the command that assures mutual responsiveness at all levels, particularly at the top.

B. Compatibility

While specific system organizations and the operational programs are properly a responsibility of the command, the hardware and languages used are subject to standardization and, in fact, must be standardized and compatible to some degree or complete chaos will result when different commands have occasion to work together. It is the responsibility of the DDR&E and JCS to ensure that commands have adequate guidance in matters of compatibility and that standards are correctly interpreted and followed.

Systems integration is concerned with the resolution of two forms of incompatibility: technical and functional. Technical incompatibility we define as language and equipment incompatibility within a system or a complex of systems. Functional incompatibility arises because of conflicts in procedure, e.g., the SAC exit problem which needed procedure coordination between SAC and NORAD.

While individual automation efforts can accomplish much, the trend toward Unified Commands and Joint Task Forces emphasizes the need for compatible inter-Service data systems. Compatibility and coordination among and between the Services is creating problems; e.g., the language incompatibility between the Army and Navy data processing systems currently precludes a completely automated fire plan for joint operations; the incompatibility between the Navy Tactical Data System and SAGE precludes convenient target information exchange between the systems. Also, few of the automation planners we talked to had given consideration to how they might exchange information with the DOD Damage Assessment Center. Obviously, the leadership for such standardization must come from the DOD level.

The majority of the command systems that were reviewed by the study group were using an interim data processing capability, or were operating manually and planning to obtain an automated capability in the future. At the present time there are few commands that depend extensively on computer aids to support the commander. The small degree of implementation of existing and proposed improvements of the command systems indicates that a large measure of compatibility can be accomplished without excessive modification of scheduling or system design parameters -- if positive action is taken in the near future.

Functional compatibility problems are more difficult to see and predict and are less amenable to standardization. Force exercises such as operation HIGH HEELS are presently the most effective method of exposing such problems. In the future, computer simulation models may become a powerful tool in such exercises.

C. Simulation Systems and Models

1. Uses of Models and Simulation. The present manual and semi-automated command information systems are exercised by the use of Command Post Exercises (CPX's). CPX's are used to train personnel, to test capability and show where improvements are needed, and to test new concepts of operation. A CPX is a simulation of operations and as such involves the use of models of the real world.

Over the past ten years the means to exercise a manual system and to use the exercises to improve the system has developed into a sophisticated technology. This technology is also currently being used extensively in computer-based operational systems. It is a semi-automated technology tailored to specific systems in its detailed application, but generalizable to a broad class of systems.

Experience with simulation vehicles demonstrates that their use -- including the specification of exercise problems and much of the interpretation of results -- is a very natural extension from similar CPX experience for operating command personnel. Simulation constitutes a major tool that can be used for improving technical capability in a command.

The simulation vehicle provides an objective, integral means by which the command can examine itself, evaluate its performance capabilities, investigate and prove out changes in structure and procedures, etc. To be fully effectual, the simulation vehicle must be used in conjunction with a continuous process of analysis of system objectives and development of performance criteria. The simulation system itself must evolve in parallel with the command information system in order to remain well-adapted to exercising it, evaluating it and verifying design changes introduced into it.

2. Dynamic Gaming. In some quarters it is hoped that a *dynamic* gaming-modeling capability can be provided for commands. Any extensive dynamic capability is certainly far off. In the current state-of-the-art, on-line gaming has only extremely limited application and is highly specialized, particularly, (1) in being restricted to quite short-range looks into the future, and (2) in using a very approximative treatment.

Immediate future prospects for the use of dynamic gaming capability by operating commands seems to be limited to two matters: (1) the provision of the facility to rapidly assess support capabilities, given a concept of approach to crisis problems; (2) generation of a limited range of contingent modifications to established plans. These sorts of capabilities, although limited, may be very important, since a major fact about the high-level command problem is that one never knows very explicitly what "game" he is playing. One can project some classes of situations that can arise and draw policy implications for response to each class -- yet in every actual crisis, the chips go down and policy changes in ways that cannot be envisioned beforehand. Consequently, any pre-planning that has been done on the basis of the defined classes of situations stands an excellent chance of not being directly applicable. In these circumstances, it is useful to provide a "rapid capability assessment" and "contingent plans modification" kind of mechanism.

There is one exception to these remarks about near-term use of dynamic gaming. There are occasionally combinations of concept and situation of unusual clarity, e.g., the SAC problem under the spasm war concept, such that a quite complete dynamic gaming approach to mission planning can be developed and can prove very effective. Unfortunately, such cases are infrequent.

3. Model Construction. The problem of constructing models (or games) has three aspects:

1. The construction of models of the system and its environment. (In the case of exercises such as CPX's, the real system may be used in a simulated environment.)
2. The provision of a mechanism of input preparation.
3. The provision of a mechanism for data reduction and analysis.

The last two aspects usually constitute the major reasons for the long time required to get results from gaming and modeling techniques. They are, in this respect, much bigger problems than the problem of constructing simulation models *per se*. It is just as vital to semi-automate these processes as it is to automate the actual models. At the present time, only in the case of simulation vehicles designed for system training have these processes been accorded the attention necessary to meet the needs of operational command.

With respect to system models themselves, a major problem is the need for flexible structure so that many different problems can be examined through facile modification of the model. Such a modification capability is to be preferred over the time-consuming construction of new models from scratch in the meticulous detail required. Techniques promising to improve radically our capability in this respect -- techniques that will provide for modular design of simulation models -- are just now emerging from the programming research laboratories.

Achieving flexibility is not only a matter of "building-block" model structure, but also a matter of data availability. We need not only information characterizing the current system, but also information about projected and hypothetical states of affairs or the means for readily generating such information.

D. Information System Analysis

Any command system is an information system whether or not it employs automation. Only from a competent over-all analysis of the existing system can it be determined whether automation can be applied effectively to any particular command problem, and such an analysis must have priority over the more detailed automation design considerations.

It is not possible to say what an optimal information system design is, in the sense of describing a firm structure.

Nonetheless, it is clear that any organization that approaches optimality will be one in which flexibility and adaptive growth are paramount. This is so since the problems the system must handle will change because the command is imbedded in a changing strategic context, has a role and a mission that may change, directs changing forces, is subject to modification of coordination requirements with lateral commands and up and down the command line, and even requires different types of man-machine functioning in different battle phases.

Both centralized and decentralized processing may be desirable, depending on the situation. Under the condition that the processing load is light, the communication net is undamaged, and the battle is in an early phase, centralized processing is most effective. To a major extent this is true because the higher level commander has a need to be responsive to particular low level events individually, in order to develop his picture of the over-all situation. Small events can be critical indicators. When the battle is joined, and the load becomes substantial or the communication is damaged, centralized control is no longer effective or desirable since the higher level commander's need then is to deal with the battle in an aggregated fashion, but on a timely basis, and centralization of detail would prevent this.

It is particularly important that information systems retain flexible patterns of usage when automation is introduced. Automation tends to reduce the variety of paths of information flow that can be employed unless flexibility is specifically provided for in the design. Tight, minimal designs should be avoided.

A good information system analysis clarifies the informational relationships among staff elements and provides the context within which various staff functions can be automated and yet remain well-integrated parts of the total command system. The goal is to develop a time-phased plan which automates one function after another, always keeping functionally integrated the over-all information system into which the parts fit.

The information system analysis can usually be expected to imply that automation should proceed in steps to maximize its benefits. In this manner, automation and thus computer programming will be applied first to those areas where the ease of problem formulation has been weighed in comparison with the system benefits to be gained. The more difficult areas from a formulation standpoint may be tackled later when the command personnel have become more familiar with the capabilities, limitations, and difficulties of automation and can apply this knowledge to sharpening the definition of the problem.

E. Security Aspects of Information Systems

1. Administrative Constraints

Every commander has certain administrative restrictions on data and information flow within his own organization as well as to external organizations. These restrictions may be initiated by the commander or by organizations external to the command.

A commander, through the aid of his staff, has close control over the type of data, the volume of data, and in general, the information that will be transferred from his command to other organizations or commands. Administrative restrictions are developed by the commander to insure that information concerning the operation or status of his command is restricted to those organizations and commands that have a need for the information.

Other administrative restrictions on information handling and distribution are issued to a commander by the Security and Intelligence communities. Procedures to protect the security of forces, the security of intelligence knowledge and intelligence sources are rigidly enforced. Data or information is available to fewer and fewer organizations and people as the sensitivity or security classification of the data increases. In any case, a need-to-know is verified before distribution of this class of information is authorized.

2. <u>The Effect of Computers on the Dissemination of Information</u>

At present, a commander is able to control the dissemination of information because elements of his command are collecting and analyzing data, generating reports, specifying security classification, initiating the distribution of information -- all manually.

The integration of computers into the command structure is expected to modify the manual operating functions of the commander's staff. Operational procedures will change, but the commander will have to retain control over the dissemination of data and information. A significant problem will be to limit segments of the files to certain users and still obtain efficient over-all file maintenance techniques within the computing facility.

The computer system data files will contain the source records that assist the Unified commander and his component commanders in carrying out their command responsibilities.

File organization within the computing system will have to reflect the administrative restrictions of the commanders.

A commander and his staff may relax administrative control in some areas, but intelligence data and other sensitive information must be physically secured on a need-to-know basis. Actually, the aggregate files within the computer would be highly classified even if single records had no classification. Also, when files are automatically transferred between computational centers (under computer control) at different echelons of command, even though a secure communications link exists between the commands, there will have to be administrative restrictions that limit file access on a need-to-know basis.

The implications of the requirement for administrative control of need-to-know in an automated system are that interlocks, modularity of system organization, special programming provisions, and the like, will be required of the system design so that operations of the command can conform to this requirement. These provisions will often be specific to this requirement, not being required for any other design or operational reason. In fact they will sometimes constitute a difficulty to be surmounted in effectively meeting other system requirements.

PART IV

LANGUAGES

A. Introduction

Language has always been a problem in command and control. Many failures of command and control in the past can be traced to failures to distinguish between a plan and an order, or between an order and a command; changing an alert status to a "higher" level of readiness may decrease the ability of the force to react to the most likely threat if the commander does not fully understand the meaning of the alert status language.

The application of computers to command and control introduces new languages. Automated systems involve the use of programming language and machine (computer) language in addition to natural English and its derivatives -- command language and design language. Unfortunately, these new languages each consist of a wide variety of tongues and dialects.

The proliferation of languages and dialects has led to the formation of several groups and committees, whose purpose is to bring some semblance of order out of chaos.

Four main considerations hamper these attempts at standardization:

1. Cost in retraining of personnel, modification of hardware, and reprogramming that would be necessary to convert to a standard.
2. In the area of programming languages there is a feeling in some quarters that data processing is such a young profession that it would do more harm than good to standardize at this point in time.
3. The language needs of every group are unique and influenced by personal preference, tradition, and inertia, so that any broad standard must of necessity be a compromise.
4. Several independent activities have arisen, each claiming jurisdiction over somewhat overlapping areas so that it frequently seems that they are contributing more to the confusion than they are to its alleviation.

A language involves an arbitrary set of rules agreed upon by both a sender (speaker, signaller) and a receiver (listener, observer). These rules, which may not be formally stated, are concerned with the meaning that the receiver should attach to some specific physical action on the part of the sender. These physical actions range from simple signals ("one if by land and two if by sea") to the very complex vocal patterns involved in human speech. It should be clear that standardization (between at least two people) is implicit in the concept of a language. The problem of standardization is not whether to standardize but instead is concerned with how widespread the standard should be and the type of information that is to be communicated.

Only recently has it become apparent that problems in areas such as inter- and intra-system compatibility are mainly problems of language. If we think of language as being a means of conveying information from one component of a system to another, to be effective, a message must be "understood" by the receiver. If this is not the case, a third component must be inserted between the sender and the receiver in order to "convert" or "translate" the message. Many names (liaison officer, interpreter, code clerk, assembler, compiler) are used to describe this third component depending upon the nature of the sender and receiver as well as whether the translation is being done by men or machines.

In the design, implementation and use of command and control systems, we are concerned with five different types of languages. These are natural, command, design, programming and machine languages. The following figure shows the various combinations of sender and receiver and the type of language used.

B. Command Language

A command language is a language used by the commander and his staff to carry on communications within the staff, with other commands (both vertical and lateral), and, if data processing equipment is involved, with the computer.

The ideal command language would be one which is sufficiently close to the commander's natural language that it would require a minimum amount of training on the part of the commander and his staff to use it, but at the same time it would be sufficiently precise in its syntax that it would be readily adaptable into a program language. In addition, an ideal command language would be economical in word usage and yet sufficiently flexible to meet the requirements of a considerable variety of information inputs and outputs imposed by the command decision function.

The command language may be characterized as a special version of English. Its semantic rules, insofar as they

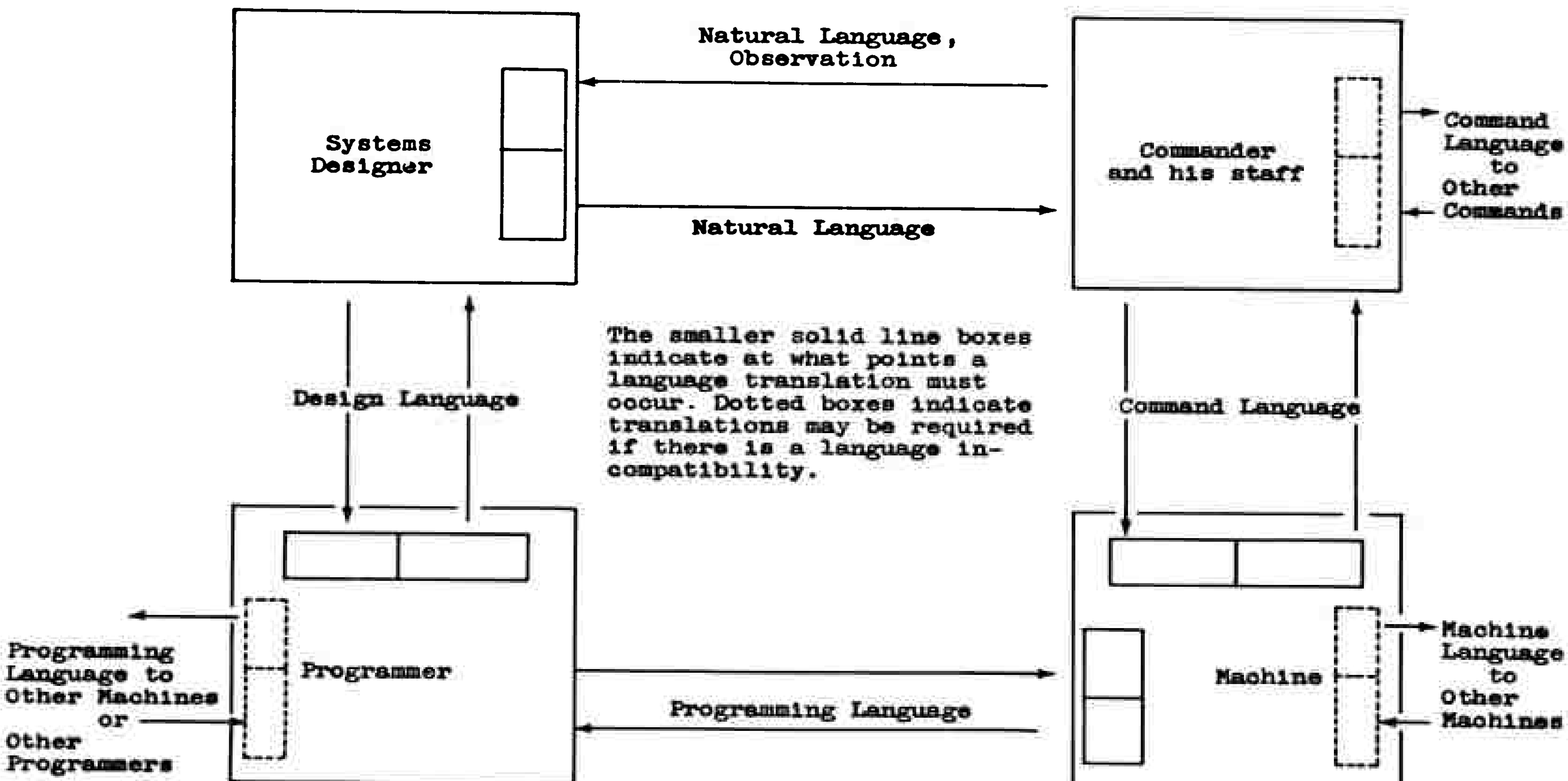
Systems Designer
Natural Language, Observation
Natural Language
Commander and his staff
Command Language to Other Commands
Design Language
The smaller solid line boxes indicate at what points a language translation must occur. Dotted boxes indicate translations may be required if there is a language incompatibility.
Command Language
Programmer
Programming Language to Other Machines or Other Programmers
Programming Language
Machine
Machine Language to Other Machines

Figure 1

differ from ordinary conversational English, are not particularly difficult for humans to understand. This modified English structure, however, is a rather complex matter involving most of the ambiguity and imprecision of English.

Communication within the command staff presents no particular problems aside from the necessity of familiarizing a new member to the nuances of the command's language.

The problems increase considerably when communication is required between commands, especially if the commands belong to different services. It hardly seems necessary to belabor the point that each of the Armed Services (and frequently each branch) has a different way of expressing the same message. The traditional solution to inter-command language problems is to appoint liaison personnel whose primary function is to translate from one command language to another. Notable failures of liaison, too numerous to mention, would lead one to suspect that military communiques, like poetry, usually lose something in the translation.

The introduction of automatic data processing into a command creates problems of a more severe nature. A computer can be programmed to understand a language only if the language can be defined unambiguously and precisely. It is mandatory that complete, precise, non-ambiguous language to developed for communication of information within a complex of men, hardware, and software. It should be obvious that such a development would be highly desirable whether or not computers are involved in the system, but it becomes absolutely necessary to the design of effective computer-based systems.

There is nothing inherent in present computer technology that precludes the automatic manipulation of language and concepts at the higher command levels. The difficulty is entirely our inability to precisely specify the language, rigorously define the concepts, and accurately describe the manipulations. It is because of this that attempts at standardization in the area of command languages have been sporadic and on the whole not very successful. However, it may be possible to devise a subset of the command language which falls within tolerable limits of these requirements, particularly if the attitude is taken that this subset would not be expected to meet <u>all</u> of the information requirements of a command system. In other words, there must always be some allowance for the conveyance of meaning by the use of the more imprecise natural language.

One advantage of having even this limited precise command language is that it forces the user to be more precise in the conveyance of information, plans, orders, and commands.

Research currently under way in other areas (mainly in information retrieval and natural language translation) will certainly aid in our understanding of command language. However, until results are received from detailed studies of the structure and function of command language, particularly as it relates to the functions of the command, there is little advantage in attempting any broad standardization; in fact, it may be better to <u>know</u> that we are speaking different languages than to <u>erroneously believe</u> we are speaking the same one.

C. <u>Design Language</u>

A design language is used by a system designer to state explicitly the functional requirements for each component of the system. It is thus differentiated from equipment specification, procedure codification, and programming language which specify how the functional requirements are to be implemented. It mediates, for example, between command language and programming language. It is a semi-technical language, but at present not formalized nor regularized in its structure. Consequently, its use depends almost entirely on the experience of the designer, and it cannot very well be taught in the sense of formal instruction, but must be learned on the job.

Part of command language, particularly the part concerned with man-machine communication, must be included in design language in the sense that design language must talk about command language in ways that concern intimate details not only of the command language structure but also of its use. As automated information processing advances to more sophisticated assistance to operational commands, the requirements on communication between command and designer communications and on their languages will become heavier.

Many current developments in the area of programming languages are attempts to raise their level to more closely approach design language, while others are attempts at formalizing design languages. Given time (about 10 years), and money, the two languages can be brought into such close correspondence that the formal language effectively also serves as the programming language. It should not be expected that this can be achieved by any sort of dramatic breakthrough but can only come about by many small advances on many fronts.

Projects aimed at raising the level of programming languages have advanced at a fairly respectable rate (and should certainly be encouraged and supported when necessary), but the more pressing problem of formalizing design languages has been almost completely neglected.

D. Programming Language

The level of programming language is currently being raised to remove concern with the explicit characteristics of the data processing equipment. This programming language, though highly stylized, makes substantial use of familiar symbolism in a way which is semi-readable and relatively easy to learn.

A programming language is used by the programmer to specify a step-by-step procedure that, when followed, will satisfy the system requirements. If the language is other than machine language, a special program is required to translate from the programming language to machine language. At a level very close to machine language we have symbolic assembly programs (SAP). At a level more closely approaching a design language, the translator is referred to as a compiler.

A multitude of higher level programming languages and compilers for their translation to machine language exist or are under development. There are currently three independent committees working at standardization. These are:

1. The ALGOL Committee of the Association for Computing Machinery. ALGOL (Algorithmic Language) was developed by representatives of data processing societies from various countries. The intent of this activity was to establish a common international programming language for scientific and engineering calculations. How well the goal of commonness has been achieved has been severely questioned. Only one group is still making a claim to implementing "full ALGOL." Most groups choose some subset of the language to which they add features that they consider desirable. There is even a movement aimed at defining a standard subset of ALGOL for use on medium size computers. To further confuse the issue, there are really two ALGOL's. ALGOL '58 was modified in 1960 to become ALGOL '60. For practical purposes the modifications were slight, but sufficient to make the two versions incompatible. ALGOL '60 today is more and more being considered as a communication

and publication language for programmers rather than a programming language as such.

2. <u>The COBOL Maintenance Committee of the Conference on Data Systems Languages (CODASYL)</u>. A meeting was called by DOD in 1959 to assess the possibility of developing a single programming language for business data processing. At that time, IBM was developing COMTRAN (Commercial Translator), Remington-Rand had FLOWMATIC, Honeywell was developing FACT, etc. At this meeting CODASYL was organized to immediately develop a common language (COBOL - Common Business Oriented Language). Perhaps more important was the self-assumed responsibility to carry on continuing research and development (R&D) into programming and system analysis techniques for business data processing. Contrary to the reports that appear from time to time, CODASYL is <u>NOT</u> sponsored by DOD. All costs (manpower, offices, meeting places, mailing, etc.) are met by voluntary donations from the participating organizations.

On the whole, COBOL comes much closer to achieving its goal than ALGOL. Although there have been two versions of COBOL - '60 and '61 - an orderly transition was made and great care was used to insure compatibility. The burgeoning of dialects prevalent in ALGOL has not occurred, mainly because a sharp line has been drawn as to what constitutes "Basic COBOL," e.g., that portion of COBOL that must be accepted by a compiler. Furthermore, there was a clear understanding that all of what is currently defined as "optional COBOL" would gradually become part of "Basic COBOL."

The R&D efforts have not been nearly as successful, primarily because of lack of support. Computer manufacturers, all of whom had a large vested interest in the design of compilers, were eager to contribute full time personnel to the COBOL effort. The same was not true for R&D, as witnessed by the fact that the COBOL designers met for a full week as often as every other week while the design was in progress, whereas the R&D groups meet for perhaps 2-3 days every other month.

3. X-3 Committee of the American Standards Association. One of its subcommittees, which is entitled "Programming Languages," has made suggestions to both the ALGOL and COBOL groups that it was the proper body to be setting and maintaining future standards in programming languages. The reply of both groups has been strongly negative. It is presently unclear what function or what influence this group will have in regard to programming languages.

Programming language for command and control applications should have, as a minimum, the following features:

1. Be machine independent, i.e., be capable of being translated to a wide variety of computers;
2. Be capable of use in formulating the broad spectrum of problems encountered in command and control systems;
3. Contain features well suited to integration and documentation of large scale systems.

ALGOL and COBOL do not meet these requirements, having been designed, respectively, for scientific and engineering calculations and for business data processing applications.

There are three programming language developments currently well advanced that are appropriate for command and control systems. They are shown in the Table below:

LANGUAGE	SYSTEM	COMPILER FOR*
CL-2 (Technical Operations, Inc.)	AIR FORCE	
	Project OMEGA	IBM 7090
JOVIAL (System Development Corp.)	DOD	
	DODDAC	CDC 1604
	ARPA Command System Research	AN/FSQ-32V (IBM)
	AIR FORCE	
	SAC Planning System	IBM 7090
	SACCS (465L)	AN/FSQ-31V (IBM)
	SPADATS	Philco 2000
	NORAD (425L)	Machine to be selected
	ESD Test Facility	IBM 7090
	NAVY	
	SPASUR	IBM 7090
NELLIAC (Naval Electronics Laboratory)	NAVY	
	NTDS	Remington Rand NTDS Computer
	ARMY	
	ADPS	Philco Basicpac
	AEPG Test Facility	IBM 709

*NELLIAC Compilers also exist for CDC 1604, IBM 704, and Burroughs 220. A JOVIAL Compiler also exists for the AN/FSQ-7 (SAGE) Computer.

Consolidation, at the current level of programming languages, is no longer a research problem. It could be and should be accomplished. The research problem is that of seeking the next level of capability in programming language, generally described as the problem-oriented language level. Far from being impeded, this research would be greatly aided by having a consolidated base from which to build.

Since neither ALGOL nor COBOL alone is sufficient in scope for command and control system programming, there appear to be three major alternatives for DOD:

1. Wait for a de facto "standard" to emerge. This seems, however, to be a way to insure further multiplicity of languages.

2. Develop a language which is essentially a blend of ALGOL and COBOL with some additional features. This would accomplish the purpose, but in doing so would create a monster which would be neither fish nor fowl. In addition to being incompatible with both ALGOL and COBOL, it would further be incompatible with current command and control programming languages.

3. Develop a language which would be a blending of CL-2 and JOVIAL and NELLIAC. This appears to be the most feasible of all alternatives for several reasons:

   (a) All three are offshoots of ALGOL and have adopted similar ways of solving many of the same problems.

   (b) All three are used mainly in military systems and could therefore be much more easily controlled by DOD, and

   (c) The cost of changes to presently existing compilers would not be prohibitive (perhaps 10 to 20 man-years total, not including conversion costs of programs other than the compilers).

E. Machine Language

Machine language is the only language that a computer understands directly. Computers can be programmed to appear to understand another languate (it is a moot point as to whether the computer or the program is doing the understanding), but it can never be "taught" to understand.

By way of analogy -- a moron can be taught to understand simple instructions in the operation of a desk calculator. We can then present him with a list of instructions (a procedure) for the solution of a complicated set of differential equations, and he can then operate the calculator to solve the problem. He has not learned to solve differential equations, instead we may only say he has been "procedurized."

A computer can be built to understand instructions in a machine language. We can then present it with a list of instructions (a program) for the solution of a complicated set of differential equations, and then the computer can be operated to solve the problem. The computer has not learned to solve differential equations, instead we may only say it has been programmed.

Almost every series of computers has had its own machine language. Where there is no requirement for communication with other computers, differences in machine language can be tolerated.

As soon as there is a necessity to communicate between computers that have different machine languages, three major problem areas arise (aside from hardware problems, such as voltage levels, transmission rate, pulsewidth, etc.). These are:

1. Instruction Lists. An instruction list is the repertoire of instructions built into the hardware of the computer. This is the area in which incompatibility is the most difficult to resolve, and fortunately, the least important as far as the ability to communicate is concerned. The only circumstance under which compatibility would be required is when actual programs rather than data are to be transmitted. For example, the requirement that all Fieldata computers have effectively the same instruction list stems in part from an early concept in the Army's Automatic Data Processing System (ADPS) concerning the possibility of a computer requesting and receiving a seldom used program from a central source whenever needed.

There are, of course, other reasons why a standard instruction list would be desirable, but considering the extreme reluctance of most computer manufacturers to standardize, this is not worth fighting about.

2. <u>Representation of Numbers</u>. This includes such things as word size, format of floating point numbers, complement form versus true value and sign, etc. Incompatibility in this area can make communication quite difficult and would certainly require an additional program either at the sender or receiver's end. Both the Naval Tactical Data System (NTDS) and ADPS programs have established (mutually incompatible) standards in this area.

3. <u>Character Set and Encoding</u>. The <u>character set</u> is the list of permissible characters that may be used in a language. An <u>encoding</u> is a specification of the form of representation of each character.

   For example, a character set might consist of the letters A through Z, the numerals 0 through 9, and the additional symbols + - / * ( ) , 1 $ ' and blank. The encoding in terms of binary digits might be 000001 for A, 000010 for B, ..., 011010 for Z, etc.

At first hand it might appear that standardization in this elementary area would be a simple matter -- but even here questions of cost, personal choice, and sheer inertia appear almost insurmountable. Despite this, three major attempts at standards (each incompatible with the other two) have been made. These are:

a. A joint Army, Navy, and Air Force committee agreed on a Standard Transmission Code (STC) for communication among the Services. This is the same code as the Army Fieldata Code used in ADPS.

b. A recent agreement by IBM, SHARE (704, 709, 7090 users group), and GUIDE (702 705 users group) to a set of compatible standards.

c. A subcommittee of the X-3 Committee of the American Standards Association. This group has as yet issued no report but "usually reliable sources" report that their standards will be significantly different from the other two groups.

There is currently little disagreement that some standardization in the machine language area is desperately needed, particularly within DOD. It would be most desirable if this standard were the same as that which will eventually be adopted by the business community (probably the one established by the X-3 committee). However, several things preclude such a choice. Chief among these are:

1. There is a strong bias on the part of ASA against government participation in, and support of, its activities. As witness to this fact, of the 31 organizations that were requested to attend the meeting which set up the X-3 committee, only three were from the government and none of these from the military. It should, therefore, be clear that there will be little chance for the military to influence the design of these standards.

2. There is the further problem of the relative ordering of numbers and letters in sorting data. Business practice, by tradition, has numbers precede letters, whereas in the

military the tradition is exactly the opposite. This can profoundly affect the choice of the optimal code.

The only alternative then, appears to be to establish Standard Transmission Code (STC) as a future standard for <u>all</u> data processing and communications within DOD. Reasons for having this standard apply to all data processing systems are two-fold. First, it is felt that there will eventually be a need for communication not only between command and control systems, but also between these systems and the business data processing type applications within the military. Secondly, it would allow for increased flexibility in the interchange of equipment between systems and the possibility of peak load assistance of one system to another; e.g., it may be possible to do planning on a computer other than the one being used in the command system.

Such standardization, if approached properly, need not be revolutionary and disruptive of present systems. To accomplish this, the standard should be established as a firm requirement for (1) use at all system interfaces (to a great extent this has already been done); (2) use in all future data processing and communication equipment. Considering the rate of development of new hardware and the consequent rate of obsolescence of present equipment, the new standard can be put into use fairly expeditiously in present systems in an evolutionary manner. If the necessity arises for communication before the standard has been fully implemented, additional programs or black boxes can be used (as they are presently).

F. Summary

It is evident that as we have gone from command language to machine language in our discussion, we have become more specific. We have gone from very vague knowledge to very explicit facts -- from very basic research to problems of implementation. From the point of view of standardization, it should be clear that machine languages could have been standardized years ago. Many of our present difficulties in this area arise because they were not. Programming languages, at their present level of development (ALGOL, COBOL, etc.), have probably just become ready for standardization. The next step of moving closer to design language will require much more developmental work. In design languages much research work into the development of a formal structure and precise definition is required. Command languages still require some very basic research into their structure and function, and it is presently not clear as to whether it will ever be possible to fully standardize. This does not preclude the possibility that we could, even today, formalize and standardize certain subsets of command and design language.

PART V

CURRENT TECHNOLOGY AND RESEARCH NEEDS

A. Hardware

1. Basic Computer Characteristics

The present rate of progress on basic computer characteristics -- logic design, speed and capacity -- is satisfactory for command applications. Currently available equipment can provide up to 500K operations per second and 500K words of high speed storage. There is no current need to develop basically new computers specifically for command applications.

2. Flexibility

To effectively achieve flexibility and growth potential in a command system, the same flexibility and potential for growth must be available in the hardware. A considerable aid to accomplishing this is the current trend to design a computer system as a collection of interconnected, compatible, standard modules (arithmetic and control units, memories, input-output devices) as opposed to a single rigidly integrated monolith. These modules, however, are now standard only within a particular manufacturer's line of equipment.

(There are some exceptions; e.g., the CDC 1604 can use IBM magnetic tape units.)

Development of compatible modules should be encouraged and supported by DOD, the goal being a family of interconnectible modules available as "off-the-shelf" items. This family should contain:

(a) A wide variety of types of modules, such as magnetic drums and tapes, disk memories, displays, printers, arithmetic and control units, etc.

(b) Within a particular type, a variety of modules of different speeds and capacities, e.g., magnetic core memories of 4K, 8K, 16K, and 32K word capacities, or arithmetic units with 50K, 100K and 1,000K operations per second speeds.

3. <u>Environment</u>

The physical size, weight and power requirements of present-day computers are serious limiting factors for some of the proposed computer-aided systems. Use of computers in military environments frequently requires a degree of ruggedness not present in commercial machines. Continued effort toward development of improved techniques for ruggedizing and miniaturizing computers for use in such environments is needed.

4. Dependability

There has been a large amount of experience gained in insuring reliability of commercial computer installations. These reliability programs have made extensive use of scheduled maintenance and marginal testing facilities for satisfying "mean time to failure" specifications. For a commercial computer installation it is satisfactory to require that the computer be in operation for a specified fraction of each day (or month, etc.). However, there are many military applications -- those that require that the computer must be in operation at critical times that are not known in advance -- for which "mean time to failure" or percentage of time operational are not suitable specifications. Therefore, there is a need to develop specifications which will insure that this requirement for "dependability" of computer operation in command systems is met.

There are techniques employed in computers for improving reliability and dependability, such as use of redundancy and modular construction, which permit operation with partial facilities. More effort is needed in developing these techniques for military needs, as well as in developing new techniques. Additional effort should be devoted to increasing the ease and rapidity of maintenance and repair of military computers and their associated hardware.

B. Communication Between Computer and User

In command systems using computers, the human linkage to the computers is critical. In one direction, this linkage consists of translation from command language, inserted into the machine through a console, to machine language. In the other direction, it consists of translation from machine language to displays understandable by the commander and his staff, i.e., to command language. The ideal for console and display combinations is to provide the human with information and action alternatives in a form close to his own natural language with its familiar flexibility and richness. The state-of-the-art provides rapid, easy communication only through fixed formats and quite limited vocabularies.

The current limitation on communication between computers and their users is only secondarily a problem of in/out equipment and computer programming. The two major limiting factors are: (1) inadequate understanding of how to determine the relevance of information (available to be displayed) to the problem at hand; (2) inability to formulate for the computer the problem of processing a command language.

The hardware and programming problem is one of not knowing what to build or program rather than one of not knowing how to do it.

C. Non-Hardware Techniques and Tools

1. Programming Languages and Techniques

Programming languages have improved immensely over the past few years. We have progressed from instructing the machine in its own language of binary digits, through use of symbolic address languages, to use of procedure-oriented languages.

Present day programming languages and techniques are such that computers can be programmed to carry out any process for which step-by-step procedures can be specified. Commercial computers are initially provided with a software package that usually includes a symbolic assembler and often a procedure-oriented compiler. In addition, users of commercial machines have available to them, through user's groups such as SHARE, CO-OP, and the like, libraries of completed programs on special applications. These libraries range from simple subroutines, such as square root or sine routines, to programs for more complicated problems, e.g., matrix inversion or linear programming. In the case of military-developed machines, the services have had to contract with programming organizations to provide such material.

Software packages are aimed primarily at speeding and easing the programming process. Some of the kinds of problems for which step-by-step procedures have been specified and that have been programmed, using such packages, are listed below.

(a) Controlling processes where an input-output relationship can be defined and maximizing criteria exist. (SAGE, oil refineries, inventory control, etc.)

(b) Storage and retrieval systems where a well-defined classification and indexing system and criteria for efficient retrieval exist.

(c) Simulation models where the environment to be simulated can be completely defined at a useful level of aggregation.

(d) Gaming and planning models where all the relevant factors which influence the game are identified and their interactions well understood.

One aspect common to all these applications is that they have been tailored to a particular user's problem. Generalized simulation models, generalized information and storage retrieval systems, etc. have not yet been of direct use.

2. Complex Processes

Our major shortcoming is our present lack of understanding for just those complex processes, basic to command, for which we feel computers will eventually be of most value. Some of these complex processes are listed below along with an indication of our current capabilities in the area.

(a) Problem-solving

Use of computers for problem-solving is in the laboratory stage. Limited programs have been written for proving theorems and for playing games such as chess, tic-tac-toe and checkers. The approaches that make use of heuristic, rule-of-thumb algorithms may be applicable to some command problems.

(b) Self-modifying or Optimizing Systems

Again, laboratory programs (e.g., Pandemonium, hill-climbing) have been written that appear to be applicable to substantive problems.

(c) Decision-making

Programs and algorithms are available for solving certain diagnostic decision problems -- problems that require the decision-maker to classify an event into one of a fixed number of unordered categories. Decisions of this type are central to control systems. Many command decisions, however, are of an entirely different type requiring evaluation of relative worth of different courses of action. Very little is known about the processes underlying such evaluative decisions.

(d) Information Retrieval

The outline or "public library" type of memory structure is easily handled by computers but appears to be grossly inefficient for many information retrieval problems. Work on associative (matrix) memory techniques are now under way.

(e) Pattern Recognition

Very little has been accomplished to-date except for very restricted problems, such as classification of sonar or radar signals and character recognition.

D. Analysis of Command Systems

A major barrier limiting the usefulness of computers in command systems is the relative lack of attention being given to research and analysis directed at understanding specific problems of the commands. A few examples of the specific command problems needing investigation are:

(1) How compatible is the command's present staff structure with the capabilities of computer systems?

(2) How much and what type of information are needed?

(3) How is intelligence information best stored, retrieved, processed and used in the command?

(4) How can computers aid a given command in plan development, evaluation, and modification?

Research on this type of problem has been grossly underemphasized. It should be reiterated that we are referring to research directed toward understanding the problems and improving the operations of particular organizations.

PART VI

RECOMMENDATIONS

Major recommendations are summarized here for the convenience of the reader. Supporting arguments are developed in more detail in preceding sections.

1. We recommend that a multi-faceted evolutionary approach be followed in development of command systems. This approach should have the following characteristics.

   a. The user of the system should participate in every step of the evolution. He is part of the system and cannot delegate his design responsibility. To exert his control over the evolution of his system will require an increase in the level of technical competence in the user's staff.

   b. A prerequisite for successful employment of computers in command systems is a thoroughgoing analysis and a precise understanding of the internal operational structure of each command and of inter-command relations and lines of authority.

To automate a function requires an explicit statement of the informational relationships between the function and the structure in which it is imbedded. Provision should be made for rapid updating of the analysis to meet changing conditions.

c. Automated command systems should be provided with an integral means for self-exercise, self-evaluation, and verification of design changes.

d. Responsible technical assistance in analysis, design, and implementation throughout the evolutionary process should be provided to the command.

e. Hardware for a command system should be drawn from a family of modular, compatible, general purpose equipments that can be configured for a wide range of system capacities. This family should be developed, improved as the state-of-the-art advances, and made available for off-the-shelf procurement.

f. The command should be given some computer capability early in the program. This will familiarize the command with computer capabilities and will aid in the evolutionary process.

g. The funding and procurement practices followed should recognize that an evolutionary program has no "operational cutover date" when the system phases from development to use and no "complete operational date," beyond which it ceases to evolve.

h. Since the system is evolutionary, at all stages it should have either unused capacity or quickly expandable capacity to allow for growth.

i. At every stage of evolution the value of the improvement through automation should outweigh the penalties paid for the use of the equipment.

2. We recommend that research be expanded to provide the knowledge and techniques needed to exploit computers more fully in our evolving command systems. Several areas of

potentially fruitful research and development relevant to command systems are listed below:

a. Development of improved techniques in formulation, analysis and programming.

b. Development of improved procedures and languages for communication between machines and their users.

c. Basic research directed toward increasing our understanding of such complex processes as pattern perception, concept formation and recognition, problem-solving, learning, and decision-making.

d. Research directed toward improving the dependability of computers and their associated hardware.

3. We recommend that much greater attention be given to the operational capability of the system after destruction or degradation of some of its elements, e.g., communications. A "life-boat list" -- functions that must be saved first -- should be provided. The features of the system that provide for its exercise under simulated conditions should include the capability to simulate such destruction and degradation.

4. We recommend strengthening the mechanisms within the DOD whereby technical and functional compatibility efforts are coordinated. When operational commands are provided with the means for developing automated command information systems according to their particular needs, it must be recognized that to achieve system integration, the commands must be provided clear guidance in command system planning, particularly in the area of compatibility constraints. Technical standards should be developed, established, and policed.

APPENDIX

BRIEFINGS AND SOURCE DOCUMENTS

In the course of the study, many meetings were held, reports read, and equipment viewed. Listed on the following pages are the meetings held and their subjects. A bibliography of reports read is also given. The systems reviewed are not described in detail because inclusion of such information would have classified the report.

Army Meetings

		Speakers	Subject(s)
1.	OCSigO, USASRDL, Army R&D	Mr. Silverstein, OCSigO Mr. Lipton, USASRDL Lt. Col. Cole, OCSigO Major Adkisson, R&D	Army Command and Control Information System (CCIS) FIELDATA family of equipments
2.	USAEPG and representatives of interested Army elements (CONARC, Artillery, Intelligence, etc.)	Maj. Gen. Urhane Col. Lotz Dr. Frese Mr. Shoemaker (TRW) Capt. Williams	Army tactical automatic data processing
3.	Aeronutronic Newport Beach, Calif.	Mr. Dolberg and staff	Army Tactical Operations Center (ARTOC)
4.	125th Signal Battalion	Lt. Col. Rippey	Tactical communications problems associated with command and control

Navy Meetings

1.	David Taylor Model Basin, Washington, D. C.	Dr. Davis Miss Todd	CINCPAC/CINCPACFLT operations control center and Navy sea surveillance center
2.	CINCPAC	Cdr. Turner	CINCPAC/CINCPACFLT operations control center
3.	COMASWDEFORPAC	V/Adm. Thatch	Command and control problems in anti-submarine warfare

Air Force Meetings	Speakers	Subject(s)
1. 425L, 473L Meeting in Washington, D. C.	Maj. Foulk (425L SPO) Dr. Fitzwater (SDC) Mr. Senisi (MITRE) Mr. Plante (SDC) Mr. Cahill (IBM)	425L, 473L
2. ESD meeting at Hanscom Field	Maj. Gen. Bergquist and staff Mr. Zracket (MITRE) Mr. Roberts (MITRE)	All Air Force command and control systems and their integration
3. SAMOS Subsystem I Equipment	Mr. Mohr and staff	Command and control hardware
JCS Meetings		
1. Joint Command and Control Study Group	Brig. Gen. McCollom and staff	JCS studies of command and control
2. Weapons Systems Evaluation Group (WSEG)	Mr. Golden Mr. Lewis Dr. Everett	WSEG work relating to the use of computers for command and control
DODDAC Meeting		
1. DODDAC Meeting	Mr. Torpey	DOD Damage Assessment Center

Other Briefings

Briefings were received in the Joint War Room, Navy Flag Plot, and the Air Force Command Post on the operation of these centers of command.

Reports

1. DDR&E answer to McNamara question on Command and Control.
2. JCS answer to McNamara question on Command and Control.
3. Air Force Winter Study Report (Draft).
4. Lincoln Laboratory Simplex Study Report No. 238, 13 February 1961.
5. "Command and Control" by Thornton Read, Princeton University Center of International Studies.
6. System and Equipment Reports on the Command and Control Systems reviewed.
7. M.R. Minsky, "Steps Toward Artificial Intelligence," PROCEEDINGS IRE, January 1961.
8. Strategic Objectives and Command Control Problems, Daniel Ellsberg, August 12, 1960.
9. Unified Action Armed Forces (UNAAF); JCS Pub. 2, November 1959.
10. DOD Communications Network Switching Study, IDA/RESD Study TR 61-7, May 1961.

11. The Enhancement of Naval Operations by Computers and Data Systems by P. L. Folsom, Capt., USN. Navy Technical Forum, Spring 1961, NAVEXOS P-2193.

12. "Methodology of System Design," Ruth Davis, David Taylor Model Basin ORD-REF830-38, June 1961.

13. "Computer System Concepts ...," Ruth Davis, David Taylor Model Basin Report C-1191, August 1960.

14. "Command Control Information System," USAEPG (Fort Huachuca) Draft Report, 15 March 1961.